OCR
Oxford Cambridge and RSA

An OCR endorsed teaching and learning tool

OCR A Level Computer Science

H446 – Paper 2

1

The IDE and programming basics

Unit 11

Programming techniques

PG ONLINE

Objectives

- Be familiar with the use of an IDE to develop and debug a program
- Define what is meant by an algorithm and pseudocode
- Learn how and when different data types are used
- Learn the basic arithmetic operations available in a typical programming language
- Write pseudocode solutions to simple problems

Facilities of an IDE

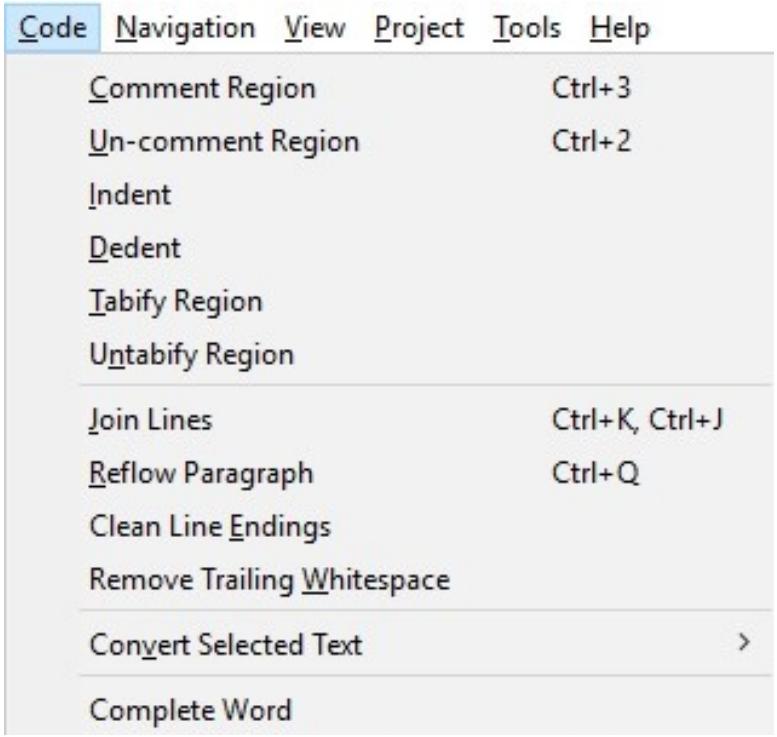
- An IDE (Integrated Development Environment) is software which enables you to enter, edit, compile (or interpret) and run your programs
- Many IDEs will also have debugging facilities to help you find the logic errors in a program
  - Why type your program in an IDE rather than a text editor?

Entering a program in an IDE

- An IDE can offer many features specifically designed to make entering program code as easy and quick as possible
  - add line numbers
  - automatically indent code
  - auto-complete commands
  - comment or un-comment a region

Typical IDE options

- Here are some of the menu options when entering program code in the IDE Komodo Edit:



Finding syntax errors

- When you interpret or compile a program, syntax errors will be reported:

```
3    a = 5
4    b = 10
5    c = 6
6    answer1 = a + b * c
7    print ("answer1 =" answer1)
8    answer2 = (a + b) * c
9    print ("answer2 =", answer2)
```

```
line 7
    print ("answer1 =" answer1)
                             ^
SyntaxError: invalid syntax
```

Finding logic errors

- Once you have corrected all the syntax errors, your program may still not run

```
3    a = 5
4    b = 10
5    c = "6"
6    answer1 = a + b * c
7    print ("answer1 =", answer1)
8    answer2 = (a + b) * c
9    print ("answer2 =", answer2)
```

```
line 6, in <module>
    answer1 = a + b * c
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Debugging facilities

- Logic errors can be hard to find
- Many IDEs provide debugging facilities
  - Set a **breakpoint** in the program which will cause the program to stop on that line
  - Set a **watch** on a variable so that its value is displayed each time it changes
  - **Step through** a program one line at a time
  - **Trace** the execution of a program – for example display a message every time a particular statement is executed or subroutine is called

Worksheet 1

- Try **Task 1** on the worksheet

Algorithms

- An algorithm is a sequence of instructions that can be followed to solve a problem
- For example, suppose you want to calculate the amount of paint needed to paint a room

Problem: calculate paint required

- What inputs would you need?
- What decisions does the painter need to make?
- Think out the sequence of instructions, starting with

  Input the dimensions of the room

- Use statements like

  Input, Calculate, Output,
  Add, Subtract, Multiply

- Or maths symbols like +, -, * (multiply)

Worksheet 1

- Complete **Task 2** of the Worksheet

Pseudocode

- Pseudocode is used to write instructions in statements that are somewhere between English and a programming language
- There are guidelines for writing pseudocode, but no strict rules
- They are an aid to thinking out the steps needed before you start to code
- Once you have written the pseudocode, the coding should be a breeze!

Pseudocode statements

- Assignment statements are written using an = sign
- The symbols +, -, *, / , ** (exponentiation) are used for common arithmetic operations

```
miles = currentMileage – previousMileage
count = count + 1
```

- Input and output statement are written

```
currentMileage = input (“Enter current mileage”)
print (“You have travelled ”, miles, “miles”)
```

Pseudocode statements

- What do the following statements do?

1. number = input (“Enter a number”)
2. number = number + 1
3. print (“The number is : ”, number)

Pseudocode

- Compare pseudocode and Visual Basic
  - What is happening?
  - What is the advantage of using pseudocode?

```
number = 0
number = input("Enter
number")

if number > 4 then
    print("Greater than 4")
else
    print("4 or less")
endif
```

```
Dim number As Integer = 0
Console.Write("Enter number")
number = Console.ReadLine()
If number > 4 Then
   Console.WriteLine("Greater than
4")
Else
   Console.WriteLine("4 or less")
End If
```

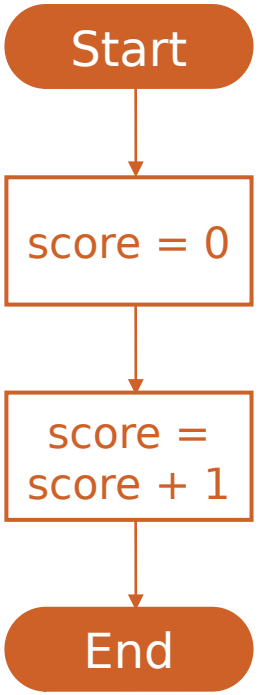
Pseudocode

- What is the purpose of pseudocode?
- What guidelines would you suggest for writing clear, understandable pseudocode?

```
number = 0
number = input ("Enter number:
")
if number > 4 then
   print ("Greater than 4")
else
   print ("4 or less")
endif
```

Variables and assignment

- An **identifier** is a name, e.g. **score**, that points to a memory location
- **Assignment** is assigning a value to the memory location
- In this example **score** is assigned the value of **0**
- In the next line, **1** is added to **score** and the result is again assigned to **score**



Data types

- Here are some variable declarations in VISUAL BASIC
  - What does each line do?

```
1.  Dim wholeNumber As Integer =
    5
2.  Dim height As Decimal = 1.5
3.  Dim name As String = "Bob"
4.  Dim choice As Char = "a"
5.  Dim validData As Boolean =
    False
```

- In Python, a variable doesn't have to be explicitly declared as a particular data type
- It happens when you assign data to the variable

Data types

- What happens if the character “9” is assigned to an integer variable?
- What are the allowable values of a Boolean data type?

```
1. Dim wholeNumber As Integer =
   5
2. Dim height As Decimal = 1.5
3. Dim name As String = "Bob"
4. Dim choice As Char = "a"
5. Dim validData As Boolean =
   False
```

Variables and Constants

- The value of a variable can be changed while the program is running
- To change the value of a constant, you have to change it in the source code and then recompile
- A constant cannot be the target of an assignment

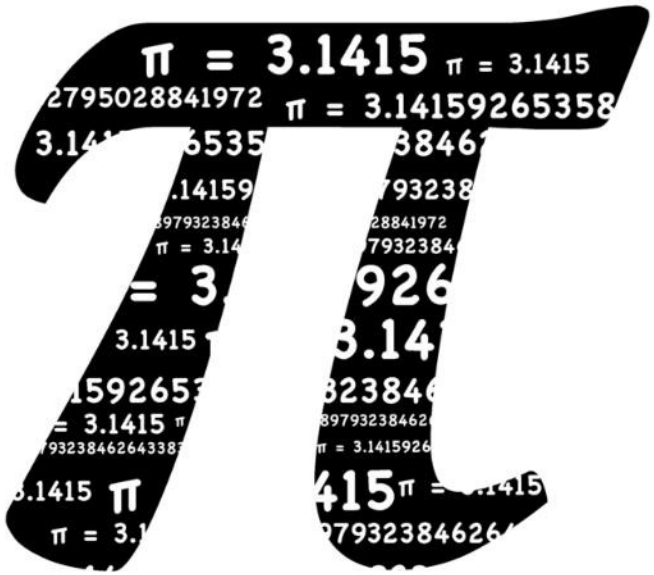
```
Dim name as String = "Robert"
Const VAT as Single = 17.5

Console.WriteLine("VAT is: " & VAT)
name = "Hazel"
Console.WriteLine("Name is: " &
name)

VAT = 20          this will cause an
error
```

Why use a constant?

- Constants reduce the risk of errors by reducing access to the memory location
  - Would you declare 3.14159265359 as a variable or a constant?



Worksheet 1

- Complete **Task 3** of the Worksheet

The mod and div operators

- The operator mod is used to find the remainder in integer division
- div returns the integer part of the division

```
x = 17 mod 3      sets x = 2
y = 17 div 3      sets y = 5
```

The mod and div operators

Calculate the following:

1. 8 mod 8
2. 8 div 8
3. 10 mod 8
4. 10 div 8

String handling functions

- Programming languages have a number of built-in string handling functions or methods
- For example:

```
name = "Robert "
x = len(name)
```

returns the length of string name

```
x = name.find("be")
```

determines if "be" occurs in string name

(returns the position of the first character in the string, 0 if not found)

String handling functions

- What would the following return? (Assume the index starts at 0)

```
name = "John"
x = len(name)
x = name.find("be")
x = name.find("oh")
```

String handling

- What values will be assigned to the variables?
  - (A string is indexed from 0)
  - What type are the variables?

```
Dim name As String = "Robert"
'Retrieves element 1 and assigns it to letter
   letter = name.Substring(1)
'Third element in the name string
   thirdChar = name(3)
'Assigns elements 1 to 3 in name to answer
   answer1 = name.Substring(1,3)
'Determines if start of string matches the search
string        answer2 = name.StartsWith("Ro")
   answer3 = name.StartsWith("Jo")
```

Worksheet 1

- Now try the questions in **Task 4** of the worksheet, and the extension task if you have time

Copyright

© 2016 PG Online Limited

The contents of this unit are protected by copyright.

This unit and all the worksheets, PowerPoint presentations, teaching guides and other associated files distributed with it are supplied to you by PG Online Limited under licence and may be used and copied by you only in accordance with the terms of the licence. Except as expressly permitted by the licence, no part of the materials distributed with this unit may be used, reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic or otherwise, without the prior written permission of PG Online Limited.

Licence agreement

This is a legal agreement between you, the end user, and PG Online Limited. This unit and all the worksheets, PowerPoint presentations, teaching guides and other associated files distributed with it is licensed, not sold, to you by PG Online Limited for use under the terms of the licence.

The materials distributed with this unit may be freely copied and used by members of a single institution on a single site only. You are not permitted to share in any way any of the materials or part of the materials with any third party, including users on another site or individuals who are members of a separate institution. You acknowledge that the materials must remain with you, the licencing institution, and no part of the materials may be transferred to another institution. You also agree not to procure, authorise, encourage, facilitate or enable any third party to reproduce these materials in whole or in part without the prior permission of PG Online Limited.

PG ONLINE